

Computer Science Department

University of Verona

A.A. 2017-18

Pattern Recognition

**Parameter estimation of a
classifier: Maximum Likelihood
and EM algorithm**

Introduzione

- Per creare un classificatore ottimale che utilizzi la regola di decisione di Bayes è necessario conoscere:
 - Le *probabilità a priori* $P(\omega_i)$
 - Le *densità condizionali* $p(\mathbf{x} | \omega_i)$
- Le performance di un classificatore dipendono fortemente dalla bontà di queste componenti

- Di solito si hanno unicamente:
 - Una *vaga conoscenza del problema*, da cui estrarre probabilità a priori.
 - *Alcuni pattern particolarmente rappresentativi, training data*, usati per *addestrare* il classificatore (spesso troppo pochi!)
- La stima delle probabilità a priori di solito non risulta particolarmente difficoltosa.
- La stima delle densità condizionali (class conditional densities) è più complessa.

- Per quanto riguarda le densità condizionali le problematiche si possono suddividere in:
 1. *Stimare la funzione sconosciuta* $p(\mathbf{x} | \omega_j)$
 2. *Stimare i parametri sconosciuti della funzione conosciuta* $p(\mathbf{x} | \omega_j)$

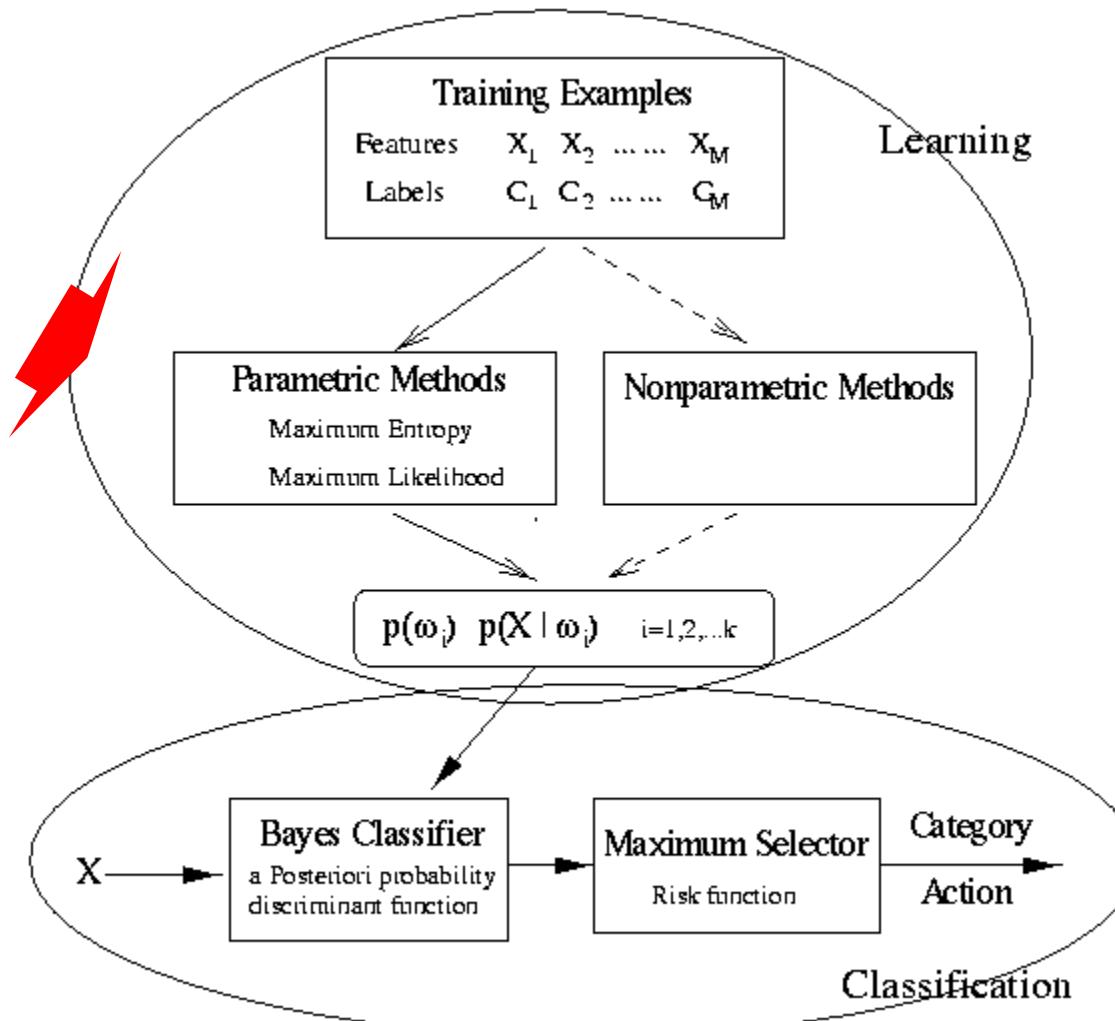
Per es., stimare il vettore $\Theta_j = (\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ se

$$p(\mathbf{x} | \omega_j) \approx N(\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$$

Stima dei parametri

- Trasferito nella *pattern recognition*, un approccio è quello di
 - 1) stimare i parametri dai dati di training
 - 2) usare le stime risultanti come se fossero valori veri
 - 3) utilizzare infine la teoria di decisione Bayesiana per usare un classificatore

Uno sguardo d'insieme



Stima dei parametri – Probabilità a priori

- Supponiamo di avere un insieme di n dati di training in cui ad ogni pattern è assegnata un'etichetta d'identità (ossia conosco per certo a quale stato ω_j appartiene il pattern k -esimo)
→ *problema di learning dei parametri supervisionato*

- Allora
$$P(\omega_i) = \frac{n_i}{n}$$

dove n_i è il numero di campioni con etichetta ω_i

Stima dei parametri – Class conditional

- Supponiamo di avere c set di campioni D_1, D_2, \dots, D_c tracciati indipendentemente in accordo alla densità $p(x|\omega_j)$
 - Assumiamo che $p(x|\omega_j)$ abbia forma parametrica conosciuta
- Il problema di stima dei parametri consiste nello stimare i parametri che definiscono $p(x|\omega_j)$
- Per semplificare il problema, assumiamo inoltre che:
 - i campioni appartenenti al set D_i non danno informazioni relative ai parametri di $p(x|\omega_j)$ se $i \neq j$.

Stima dei parametri – Due approcci

- Specificatamente, il problema può essere formulato come:
 - Dato un set di training $D=\{x_1, x_2, \dots, x_n\}$
 - $p(x|\omega)$ è determinata da θ , che è un vettore rappresentante i parametri necessari
(p.e., $\theta = (\mu, \Sigma)$ se $p(x | \omega) \approx N(\mu, \Sigma)$)
 - Vogliamo trovare il migliore θ usando il set di training.
- Esistono due approcci
 - Stima **Maximum-likelihood** (ML)
 - Stima parametrica Bayesiana

Da come output dei numeri che sono media e matrice di covarianza.

Non dà dei parametri ma delle stime di parametri con la loro probabilità

Stima dei parametri – Due approcci (2)

- Approccio Maximum Likelihood
 - I parametri sono *quantità fissate* ma sconosciute
 - La migliore stima dei loro valori è quella che *massimizza la probabilità di ottenere i dati di training*
- Approccio Bayesiano
 - I parametri sono *variabili aleatorie* aventi determinate probabilità a priori
 - Sono necessari *iperparametri*
 - Le osservazioni dei dati di training trasformano queste probabilità in probabilità a posteriori

Stima dei parametri – Due approcci (3)

- Aggiungendo campioni di training il risultato è di rifinire meglio la forma delle densità a posteriori, causando un innalzamento di esse in corrispondenza dei veri valori dei parametri (fenomeno di *Bayesian Learning*).
- I risultati dei due approcci, benché proceduralmente diversi, sono qualitativamente simili.

Approccio *Maximum Likelihood*

- In forza dell'ipotesi di partenza del problema, poiché i pattern del set \mathbf{D} sono i.i.d., abbiamo che:

$$p(\mathbf{D} \mid \boldsymbol{\theta}) = \prod_{k=1}^n p(x_k \mid \boldsymbol{\theta})$$

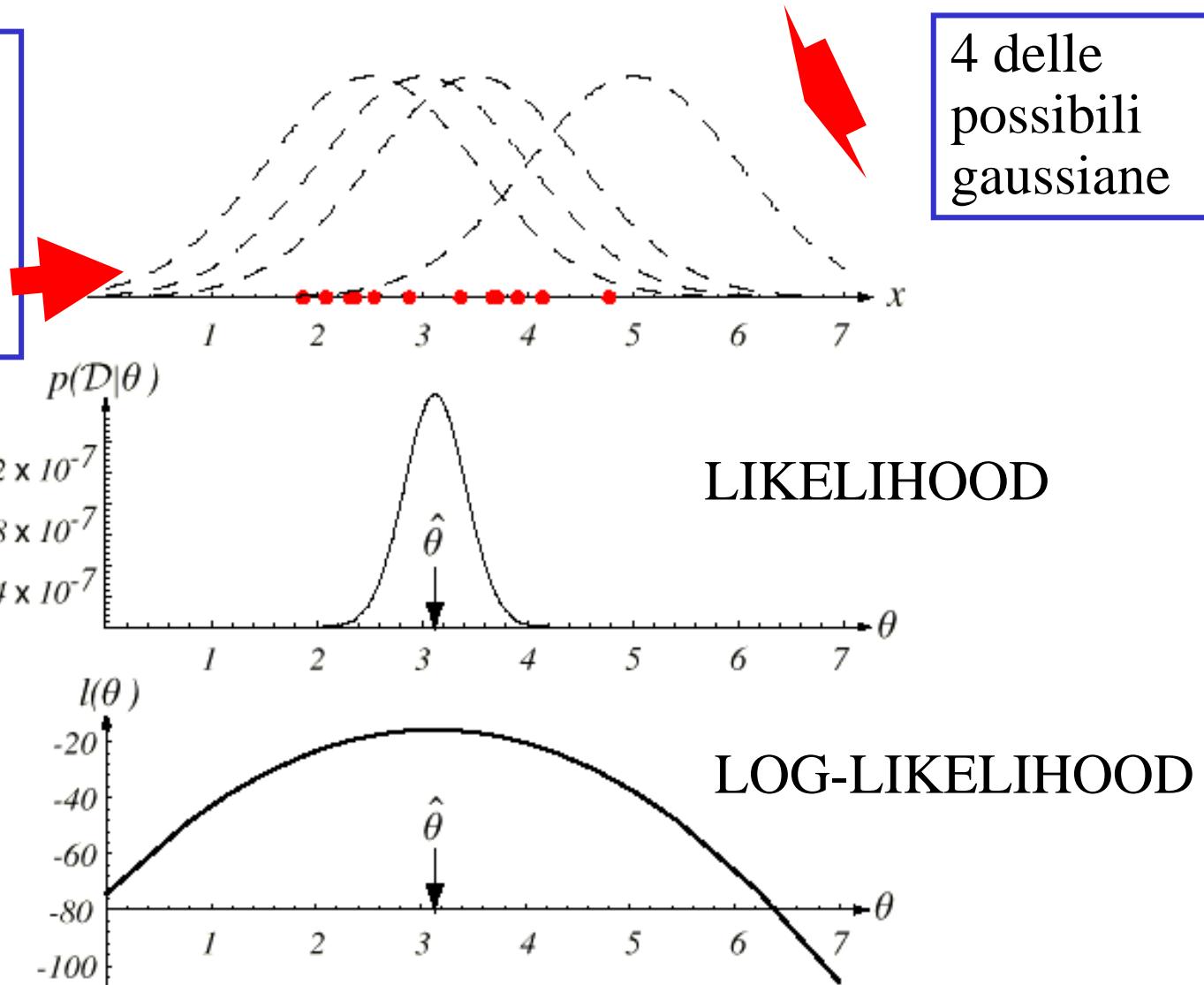
Probabilità dei miei dati di training (D) dati dei parametri (theta)

- Vista come funzione di $\boldsymbol{\theta}$, $p(\mathbf{D}|\boldsymbol{\theta})$ viene chiamata *likelihood* di $\boldsymbol{\theta}$ rispetto al set di campioni D .
- La stima di Maximum Likelihood di $\boldsymbol{\theta}$ è, per definizione, il valore $\hat{\boldsymbol{\theta}}$ che massimizza $p(\mathbf{D}|\boldsymbol{\theta})$;
- Ricordiamo l'assunzione che $\boldsymbol{\theta}$ è fissato ma sconosciuto

Approccio Maximum Likelihood (2)

Punti di training 1-D assunti generati da una densità gaussiana di varianza fissata ma media sconosciuta

4 delle possibili gaussiane



NB: La likelihood $p(D|\theta)$ è funzione di θ , mentre la densità condizionale $p(x|\theta)$ funzione di x

Approccio Maximum Likelihood (3)

- Se il numero di parametri da stimare è p , sia $\theta = (\theta_1 \dots \theta_p)^t$ e
- Per motivi matematici risulta più semplice lavorare con il logaritmo della likelihood.
- Definiamo quindi $l(\theta)$ come *funzione di log-likelihood*

$$\nabla \theta \equiv \begin{bmatrix} \frac{\partial}{\partial \theta_1} \\ \vdots \\ \frac{\partial}{\partial \theta_p} \end{bmatrix}$$

$$l(\theta) \equiv \ln p(D | \theta) = \sum_{k=1}^n \ln p(x_k | \theta)$$

Approccio Maximum Likelihood (4)

- Lo scopo è di ottenere quindi il vettore

$$\hat{\boldsymbol{\theta}} = \arg \max_{\boldsymbol{\theta}} l(\boldsymbol{\theta})$$

in cui la dipendenza sul data set D è implicita.

- Pertanto per ricavare il max:

$$l(\boldsymbol{\theta}) \equiv \ln p(\mathbf{D} | \boldsymbol{\theta}) = \sum_{k=1}^n \ln p(x_k | \boldsymbol{\theta})$$



$$\nabla_{\boldsymbol{\theta}} l(\boldsymbol{\theta}) = \sum_{k=1}^n \nabla_{\boldsymbol{\theta}} \ln p(x_k | \boldsymbol{\theta})$$

da cui vogliamo ottenere $\nabla_{\boldsymbol{\theta}} l(\boldsymbol{\theta}) = 0$

Maximum Likelihood: caso Gaussiano

- Consideriamo che i campioni siano generati da una popolazione normale multivariata di media μ e covarianza Σ .
- Per semplicità, consideriamo il caso in cui solo la media μ sia sconosciuta. Consideriamo quindi il punto campione \mathbf{x}_k e troviamo:

$$\ln p(\mathbf{x}_k | \mu) = -\frac{1}{2} \ln [(2\pi)^d |\Sigma|] - \frac{1}{2} (\mathbf{x}_k - \mu)^t \Sigma^{-1} (\mathbf{x}_k - \mu)$$



$$\nabla_{\mathbf{x}} \mathbf{x}^t \mathbf{B} \mathbf{x} = (\mathbf{B} + \mathbf{B}^t) \mathbf{x}$$

Matrix cookbook pag.11 (81)

$$\nabla_{\mu} \ln p(\mathbf{x}_k | \mu) = -\Sigma^{-1} (\mathbf{x}_k - \mu)$$

Maximum Likelihood: caso Gaussiano (2)

- Identificando θ con μ si deduce che la stima Maximum-Likelihood di μ deve soddisfare la relazione:

$$\sum_{k=1}^n -\Sigma^{-1}(\mathbf{x}_k - \hat{\boldsymbol{\mu}}) = 0$$

- Moltiplicando per Σ , cambiando il segno, e riorganizzando la somma otteniamo

$$\hat{\boldsymbol{\mu}} = \frac{1}{n} \sum_{k=1}^n \mathbf{x}_k$$

che non è altro che la semplice *media* degli esempi di training, altresì indicata con $\hat{\boldsymbol{\mu}}_n$ per indicarne la dipendenza dalla numerosità del training set.

$$\sum_{k=1}^n \Sigma^{-1} (\mathbf{x}_k - \hat{\boldsymbol{\mu}}) = 0$$

$$\Sigma^{-1} \Sigma \sum_{k=1}^n (\mathbf{x}_k - \hat{\boldsymbol{\mu}}) = 0 \times \Sigma$$

$$\sum_{k=1}^n (\mathbf{x}_k - \hat{\boldsymbol{\mu}}) = 0$$

$$(\mathbf{x}_1 - \hat{\boldsymbol{\mu}}) + (\mathbf{x}_2 - \hat{\boldsymbol{\mu}}) + \dots + (\mathbf{x}_n - \hat{\boldsymbol{\mu}}) = 0$$

$$\mathbf{x}_1 - \hat{\boldsymbol{\mu}} + \mathbf{x}_2 - \hat{\boldsymbol{\mu}} + \dots + \mathbf{x}_n - \hat{\boldsymbol{\mu}} = 0$$

$$\sum_{k=1}^n \mathbf{x}_k - n\hat{\boldsymbol{\mu}} = 0$$

$$\hat{\boldsymbol{\mu}} = \frac{1}{n} \sum_{k=1}^n \mathbf{x}_k$$

Maximum Likelihood: caso Gaussiano (3)

- Consideriamo ora il caso più tipico in cui la distribuzione Gaussiana abbia media e covarianza ignote.
- Consideriamo prima il caso univariato $\theta = (\theta_1, \theta_2) = (\mu, \sigma^2)$
- Se si prende un singolo punto x_k abbiamo

$$\ln p(x_k | \theta) = -\frac{1}{2} \ln[2\pi\theta_2] - \frac{1}{2\theta_2} (x_k - \theta_1)^2$$

la cui derivata è

$$\nabla_{\theta} l = \nabla_{\theta} \ln p(x_k | \theta) = \begin{bmatrix} \frac{1}{\theta_2} (x_k - \theta_1) \\ -\frac{1}{2\theta_2} + \frac{(x_k - \theta_1)^2}{2\theta_2^2} \end{bmatrix}$$

Maximum Likelihood: caso Gaussiano (4)

- Eguagliando a 0 e considerando tutti i punti si ottiene:

$$\sum_{k=1}^n \frac{1}{\hat{\theta}_2} (x_k - \hat{\theta}_1) = 0 \quad - \sum_{k=1}^n \frac{1}{\hat{\theta}_2} + \sum_{k=1}^n \frac{(x_k - \hat{\theta}_1)^2}{\hat{\theta}_2^2} = 0$$

dove $\hat{\theta}_1$ e $\hat{\theta}_2$ sono le stime ML per θ_1 e θ_2 .

- Sostituendo $\hat{\mu} = \hat{\theta}_1$ e $\hat{\sigma}^2 = \hat{\theta}_2$ si hanno le stime ML di media e varianza

$$\hat{\mu} = \frac{1}{n} \sum_{k=1}^n x_k \quad \hat{\sigma}^2 = \frac{1}{n} \sum_{k=1}^n (x_k - \hat{\mu})^2$$

Maximum Likelihood: caso Gaussiano (5)

- Il caso multivariato si tratta in maniera analoga con più conti. Il risultato è comunque:

$$\hat{\boldsymbol{\mu}} = \frac{1}{n} \sum_{k=1}^n \mathbf{x}_k \quad \hat{\boldsymbol{\Sigma}} = \frac{1}{n} \sum_{k=1}^n (\mathbf{x}_k - \hat{\boldsymbol{\mu}})(\mathbf{x}_k - \hat{\boldsymbol{\mu}})^t$$

- Si noti tuttavia che la stima della covarianza è sbilanciata, cioè una **sottostima** (vedasi materiale aggiuntivo)
- Per avere uno stimatore unbiased si deve correggere così:

$$\frac{1}{n} \longrightarrow \frac{1}{n-1}$$

Maximum-Likelihood: altri casi

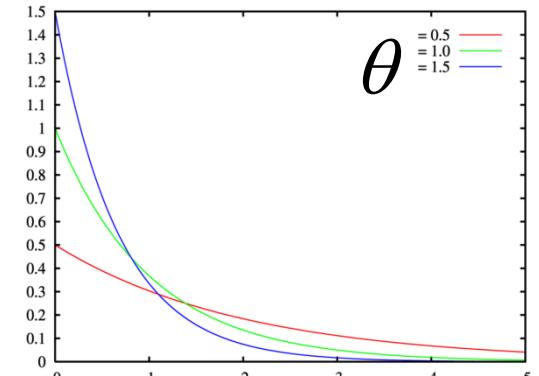
- Esistono, oltre alla densità Gaussiana, anche altre famiglie di densità che costituiscono altrettante famiglie di parametri:

- *Distribuzione esponenziale*

$$p(x | \theta) = \begin{cases} \theta e^{-\theta x} & x \geq 0 \\ 0 & \text{altrimenti} \end{cases}$$

- *Distribuzione uniforme*

$$p(x | \theta) = \begin{cases} 1/\theta & 0 \leq x \leq \theta \\ 0 & \text{altrimenti} \end{cases}$$



Maximum-Likelihood – Modello d'errore

- In generale, se i modelli parametrici sono validi, il classificatore *maximum-likelihood* fornisce risultati eccellenti.
- Invece, se si usano famiglie parametriche scorrette, o stime di parametri sbagliati (il training set non è completo e rappresentativo) il classificatore produce forti errori

Maximum-Likelihood – Modello d'errore (2)

- Di fatto *manca un modello d'errore che dia un voto alla parametrizzazione ottenuta.*
- Inoltre, per applicare la stima di Maximum-Likelihood, tutti i dati di training devono essere disponibili
 - Se vogliamo utilizzare nuovi dati di training, è necessario ricalcolare la procedura di stima Maximum-Likelihood.

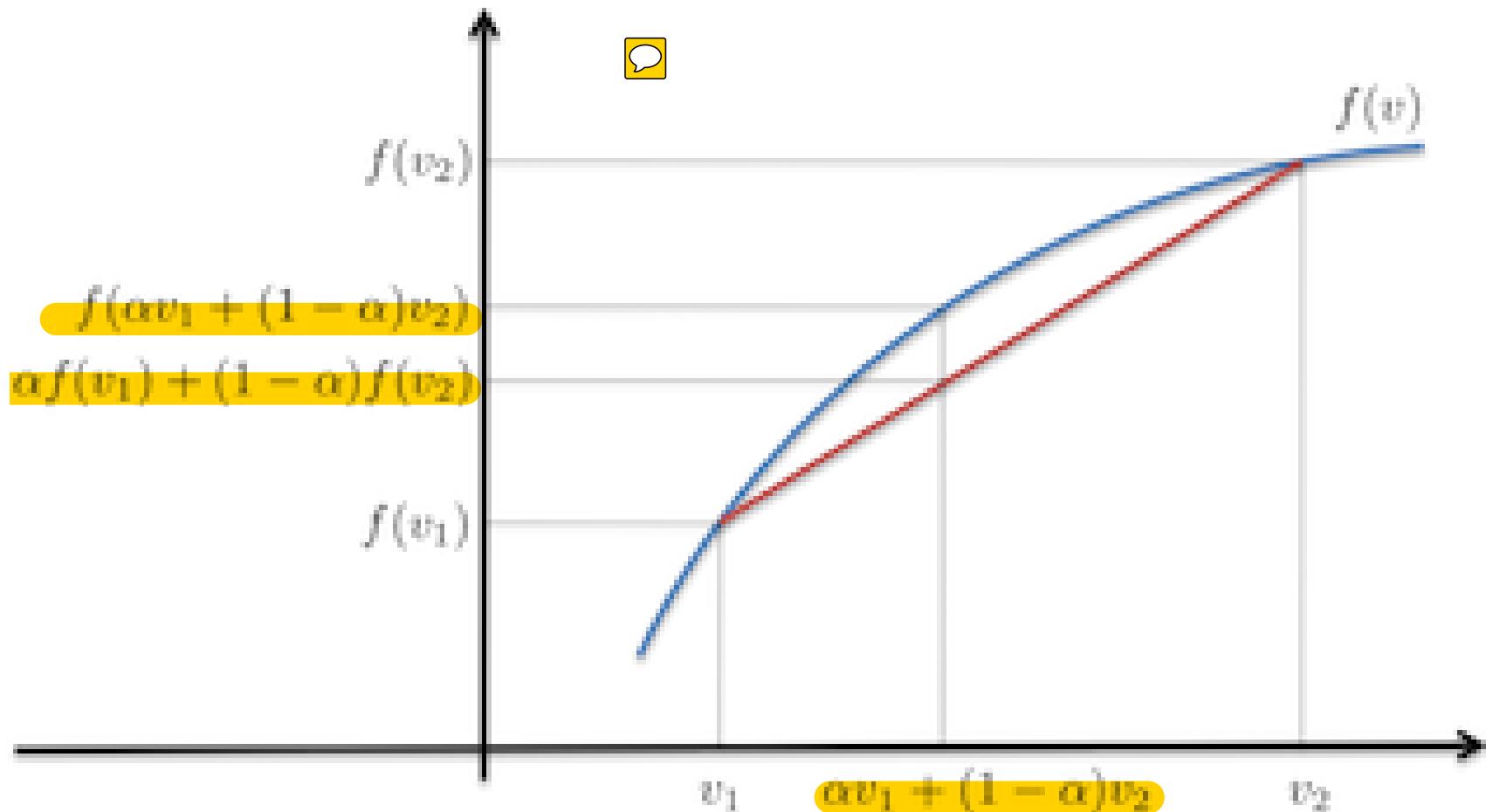
Diseguaglianza di Jensen

- La funzione $\varphi: R \rightarrow R$ si dice **concava** se per ogni coppia di numeri $x_1, x_2 \in R$ e di pesi positivi $\lambda_1, \lambda_2 \in R$ tali che $\lambda_1 + \lambda_2 = 1$ abbiamo

$$\varphi(\lambda_1 x_1 + \lambda_2 x_2) \geq \lambda_1 \varphi(x_1) + \lambda_2 \varphi(x_2)$$

- La funzione φ si dice convessa se $-\varphi$ è concava

Diseguaglianza di Jensen



Diseguaglianza di Jensen

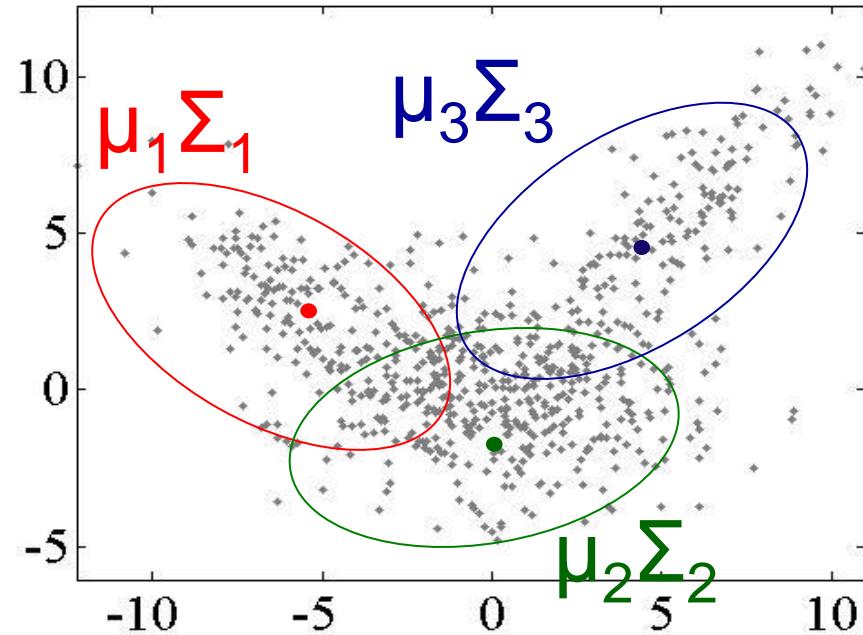
- La diseguaglianza di Jensen si generalizza a somme di n fattori

$$\varphi\left(\sum_{k=1}^n \lambda_k x_k\right) \geq \sum_{k=1}^n \varphi(\lambda_k x_k)$$

- con φ concava e

$$\sum_{k=1}^n \lambda_k = 1$$

Expectation-Maximization



$$p(v^{(t)} | h^\theta) = \sum_{i=1}^M \pi_i N(\mu_i, \Sigma_i)$$

Nel mio caso M=3



Introduction - Maximum Likelihood Estimation (MLE) problem

- INPUT:
 - A dataset of **observations** $v = \{v^{(t)}\}_{t=1 \dots T}$
 - An *implicit* knowledge, i.e.
 - the dataset comes from a parametric random process
 - such random process has a known form (f.i. a mixture of Gaussians)
 - other (i.i.d. data, usually)
- OUTPUT:
 - the **set of parameters h^θ** that maximizes the *likelihood* $p(v|h^\theta)$ a.k.a. *objective function* $L(h^\theta)$



Introduction - MLE problem and EM solution

- Usually, the MLE is performed by differentiating the likelihood function with respect to the various parameters, and solving for 0

$$\frac{\partial \log p(v | h^\theta)}{\partial h_i^\theta} = 0 \quad i = 1 \dots I$$

- Sometimes, this solution is not feasible due to the complex form of the likelihood
- This is the situation in which the EM algorithm helps

Introduction - EM

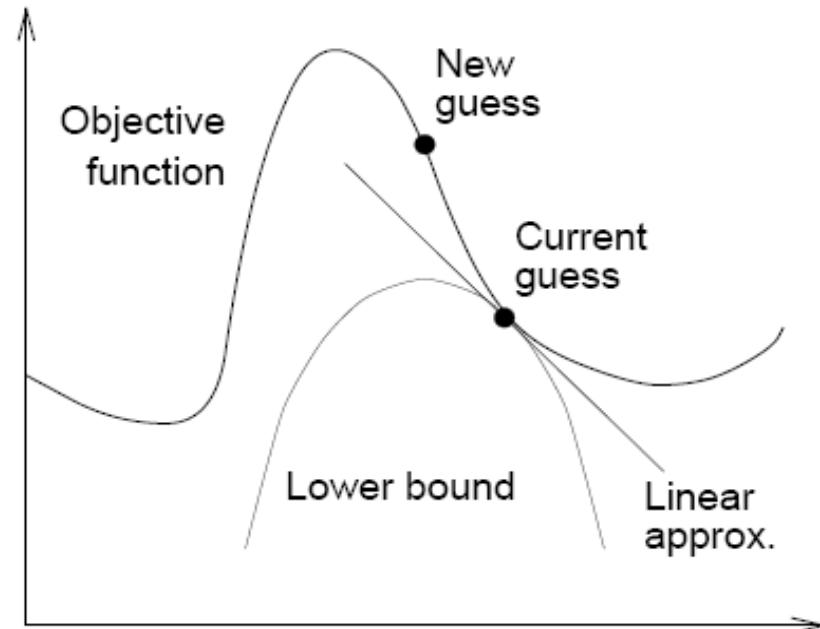


- Iterative process
- Each iteration is composed by 2 steps
 - E-step: Expectation
 - M-step: Maximization
- Convergent to a local maxima of the likelihood function
- Widespreadly used
 - genetics
 - statistics
 - econometrics

Introduction -

EM placement in the maximization methods literature

- ***Gradient descent***: linear approximation to the $L(h^\theta)$
 - we don't know how good is the approximation
 - we don't know how big the step to do
- ***Newton methods***: quadratic approx
 - same problem as above
- ***EM***:
 - at each E step it builds a local lower bound of the objective function
 - at each M step, a novel h^θ which corresponds to a bigger value of the objective function



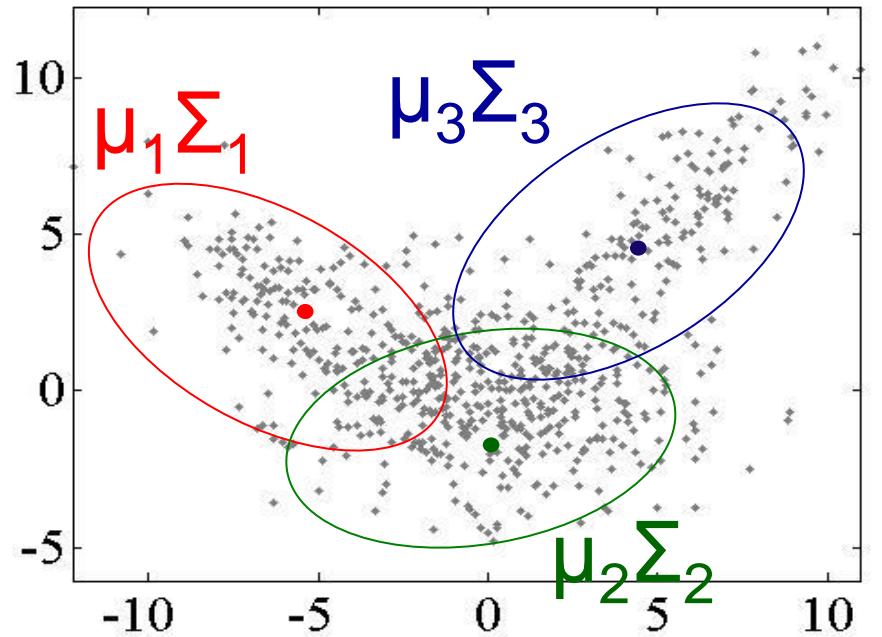
Introduction - MLE example - Mixture of Gaussians (MoG)

- $p(v^{(t)} | h^\theta) = \sum_{i=1}^M \pi_i N(\mu_i, \Sigma_i)$

- $p(v | h^\theta) = \prod_{t=1}^T p(v^{(t)} | h^\theta)$

- $\log p(v | h^\theta) = \sum_{t=1}^T \log p(v^{(t)} | h^\theta)$

- $\log p(v | h^\theta) = \sum_{t=1}^T \log \sum_{i=1}^M \pi_i N(\mu_i, \Sigma_i)$



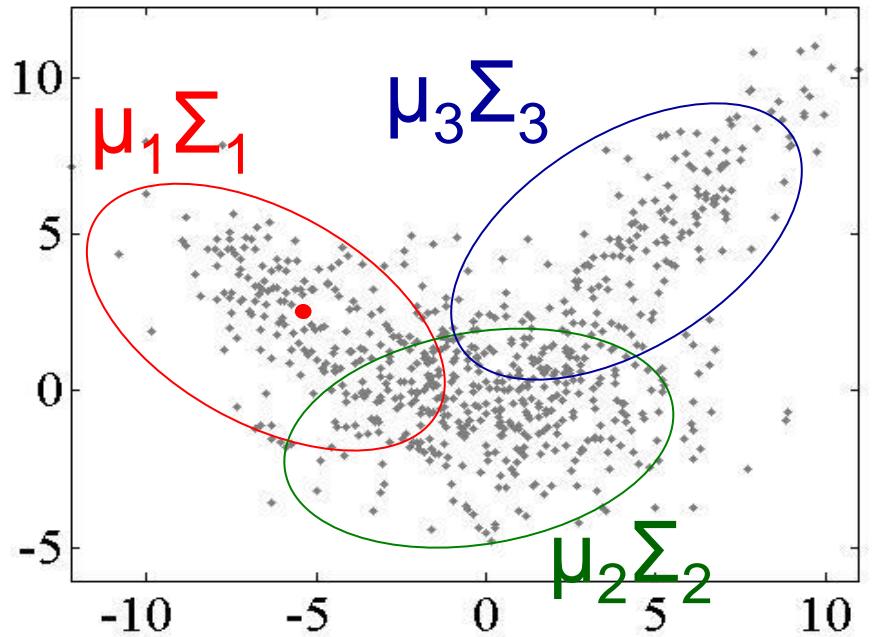
Introduction - MLE example - MoGs (2)

- $p(v^{(t)} | h^\theta) = \sum_{i=1}^M \pi_i N(\mu_i, \Sigma_i)$

- $p(v | h^\theta) = \prod_{t=1}^T p(v^{(t)} | h^\theta)$

- $\log p(v | h^\theta) = \sum_{t=1}^T \log p(v^{(t)} | h^\theta)$

- $\log p(v | h^\theta) = \sum_{t=1}^T \log \sum_{i=1}^M \pi_i N(\mu_i, \Sigma_i)$



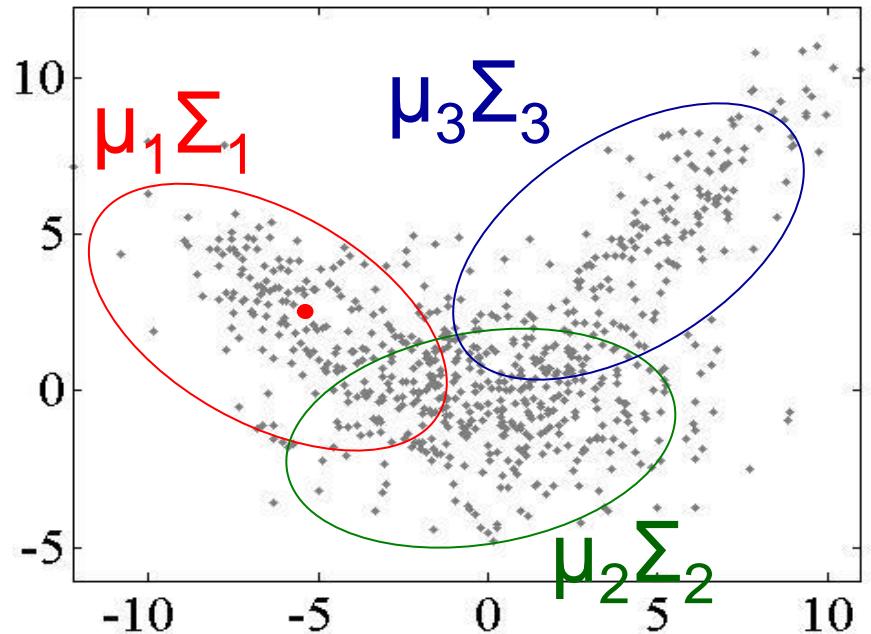
Introduction - MLE example - MoGs (3)

- $\log p(v | h^\theta) = \sum_{t=1}^T \log \sum_{i=1}^M \pi_i N(\mu_i, \Sigma_i)$

- Goals

1. find h^θ
2. maximize $\log p(v | h^\theta)$

$$\frac{\partial \log p(v | h^\theta)}{\partial h^\theta} = 0$$



PROBLEMATIC

the parameters are coupled, due to the
sum of the log: [no closed form solution](#)

The algorithm - EM in one slide! - The EM trick

$$\begin{aligned}
 \log P(v | h^\theta) &= \log P(v) \\
 &= \log \left(\int_h P(h, v) \right) \\
 &= \log \left(\int_h Q(h) \frac{P(h, v)}{Q(h)} \right) \\
 &\geq \int_h Q(h) \log \left(\frac{P(h, v)}{Q(h)} \right) = -F(Q(h), P(h, v))
 \end{aligned}$$

Divergenza: voglio confrontare quanto
Q(h) dista da P(h,v)

Jensen Inequality **The trick**

The algorithm - Novel objects in the MLE instance

$$\int_h Q(h) \log \left(\frac{P(h, v)}{Q(h)} \right) = -F(Q(h), P(h, v)) \leq \log(P(v))$$

- h = *hidden variable*
 - a hidden *quality* of the single data point
- $P(h, v)$ = *complete data (hidden + visible) likelihood*
 - it explains how the hidden variables and the visible ones are coupled together
- $Q(h)$ = *support distribution on the hidden variables*
 - a distribution over the hidden variables, simpler than $P(h, v)$

The algorithm - Novel objects in the MLE instance (2)

$$\int_h Q(h) \log \left(\frac{P(h, v)}{Q(h)} \right) = -F(Q(h), P(h, v)) \leq \log(P(v))$$

- • $F(Q(h), P(h, v))$
 - a divergence between Q, P a functional
 - an inferior bound with respect to the objective function $L(h^\theta)$
 - an object with $Q(h)$ unknown
 - an object with h^θ unknown

The algorithm - Minimization of the divergence

- I minimize $F(Q, P)$ alternatively (*the sign – vanishes during minimization*)

1. with respect to $Q(h)$, with h^θ fixed



$$\frac{\partial}{\partial Q(h)} \int_h Q(h) \log \left(\frac{P(h, v)}{Q(h)} \right) = 0$$



$$Q(h) = P(h | v, h^\theta)$$

2. with respect to h^θ , with $Q(h)$ fixed



$$\frac{\partial}{\partial h^\theta} \int_h Q(h) \log \left(\frac{P(h, v)}{Q(h)} \right) = 0$$



$$\int_h Q(h) \frac{\partial}{\partial h^\theta} \log P(h, v | h^\theta) = 0$$



The algorithm -

The core of the EM in practice

- **INITIALIZATION:** set an initial h^θ
- **STEP E:** Minimize $F(Q, P)$ with respect to $Q(h^{(t)})$ calculating for each possible value of $h^{(t)}$

$$Q(h^{(t)}) \leftarrow P(h^{(t)} | v^{(t)}, h^\theta)$$

EASY TO
COMPUTE !!!

for each t

- **STEP M:** Minimize $F(Q, P)$ with respect to h^θ solving

$$\sum_{t=1}^T \left(\int_{h^{(t)}} Q(h^{(t)}) \frac{\partial}{\partial h^\theta} \log P(h^{(t)}, v^{(t)} | h^\theta) \right) = 0$$

EASY TO
COMPUTE !!!

for M parameters, this is a system of M equations.



The algorithm - Perplexities and practical receipts

- Cool, but when should I use EM?
 - with probabilistic problems, in which mixtures of *whatever* are involved, where each data point is generated by one of the components of the mixture
 - MoG (mixtures of Gaussian)
 - HMM (mixtures of states)
 - Bayes Net (mixtures of parents of a node)
- Crucial question: what is $h^{(t)}$?
 - $h^{(t)}$ indicates what component of the mixture generates the data $v^{(t)}$

Applications - Back to the MoGs - the E-players

- $p(v^{(t)} | h^\theta) = \sum_{i=1}^M \pi_i N(\mu_i, \Sigma_i)$

- $p(v^{(t)} | h^{(t)}, h^\theta) = N(\mu_{h^{(t)}}, \Sigma_{h^{(t)}})$

- $p(h^{(t)} | h^\theta) = \pi_{h^{(t)}}$

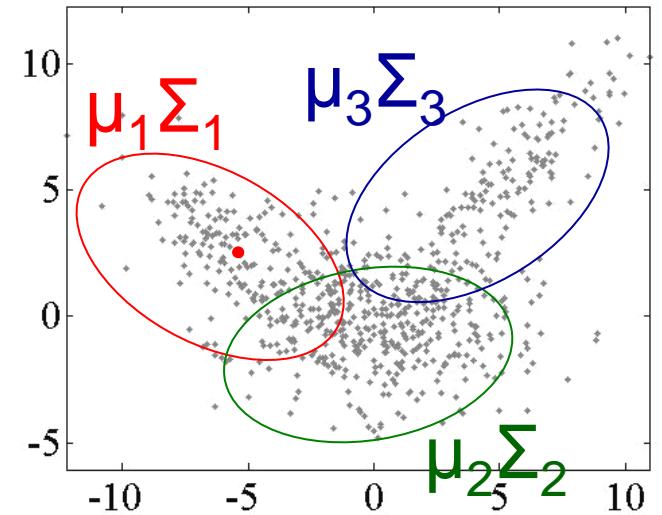
BAYES

- $P(h^{(t)} | v^{(t)}, h^\theta) = \frac{P(v^{(t)} | h^{(t)}, h^\theta) P(h^{(t)} | h^\theta)}{P(v^{(t)} | h^\theta)} = \frac{P(v^{(t)} | h^{(t)}, h^\theta) P(h^{(t)} | h^\theta)}{\sum_{h^{(t)}} P(v^{(t)} | h^{(t)}, h^\theta) P(h^{(t)} | h^\theta)}$



$$Q(h^{(t)})$$

Compute for each i, for each t



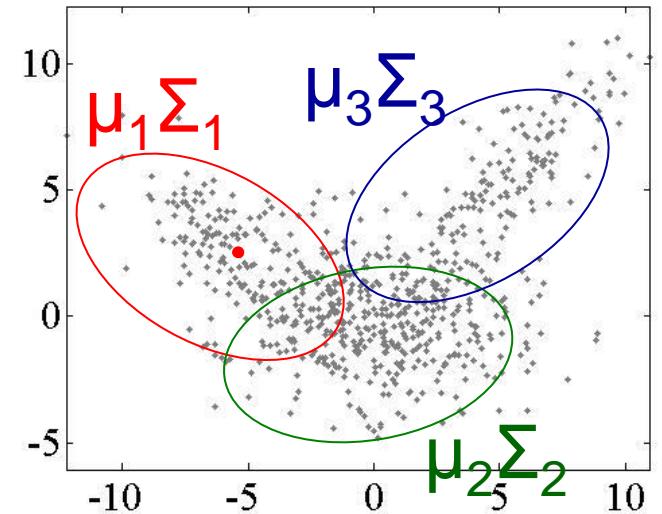
$$= \frac{\pi_{h^{(t)}} N(\mu_{h^{(t)}}, \Sigma_{h^{(t)}})}{\sum_{h^{(t)}} \pi_{h^{(t)}} N(\mu_{h^{(t)}}, \Sigma_{h^{(t)}})}$$

Applications -

Back to the MoGs - the M-players

- $\sum_{t=1}^T \left(\int Q(h^{(t)}) \frac{\partial}{\partial \hat{h}^\theta} \log P(h^{(t)}, v^{(t)} | h^\theta) \right) = 0$

$$= \sum_{t=1}^T \sum_{h^{(t)}=1}^K Q(h^{(t)}) \frac{\partial}{\partial \hat{h}^\theta} \log P(h^{(t)}, v^{(t)} | h^\theta) = 0$$



$$= \sum_{t=1}^T \sum_{h^{(t)}=1}^K$$

$$P(h^{(t)} | v^{(t)}, \hat{h}^\theta) \frac{\partial}{\partial \hat{h}^\theta} \log P(v^{(t)} | h^{(t)}, h^\theta) P(h^{(t)} | h^\theta)$$

$$\sum_{t=1}^T \sum_{h^{(t)}=1}^K P(h^{(t)} | v^{(t)}, \hat{h}^\theta) \frac{\partial}{\partial \hat{h}^\theta} \log P(v^{(t)} | h^{(t)}, h^\theta)$$

$$\sum_{t=1}^T \sum_{h^{(t)}=1}^K P(h^{(t)} | v^{(t)}, \hat{h}^\theta) \frac{\partial}{\partial \hat{h}^\theta} \log P(h^{(t)} | h^\theta)$$

!!!

$$\sum_{t=1}^T \sum_{h^{(t)}=1}^K P(h^{(t)} | v^{(t)}, \hat{h}^\theta) \frac{\partial}{\partial \hat{h}^\theta} \log P(v^{(t)} | h^{(t)}, h^\theta)$$

(ricorda! $\dashrightarrow P(v^{(t)} | h^{(t)}, h^\theta) = N(\mu_{h^{(t)}}, \Sigma_{h^{(t)}})$)

$$\sum_{t=1}^T \sum_{h^{(t)}=1}^K P(h^{(t)} | v^{(t)}, \hat{h}^\theta) \frac{\partial}{\partial \mu_k} \log N(\mu_{h^{(t)}}, \Sigma_{h^{(t)}})$$

$$= \sum_{t=1}^T P(h^{(t)} = k | v^{(t)}, \hat{h}^\theta) \Sigma^{-1} (v^{(t)} - \mu_k) = 0$$

$$= \sum_{t=1}^T P(h^{(t)} = k | v^{(t)}, \hat{h}^\theta) \Sigma^{-1} v^{(t)} - P(h^{(t)} = k | v^{(t)}, \hat{h}^\theta) \Sigma^{-1} \mu_k = 0$$

$$\mu_k = \frac{\sum_{t=1}^T P(h^{(t)} = k | v^{(t)}, \hat{h}^\theta) v^{(t)}}{\sum_{t=1}^T P(h^{(t)} = k | v^{(t)}, \hat{h}^\theta)}$$



$$\mu_k = \frac{\sum_{t=1}^T P(h^{(t)} = k \mid v^{(t)}, \hat{h}^\theta) v^{(t)}}{\sum_{k=1}^K \sum_{t=1}^T P(h^{(t)} = k \mid v^{(t)}, \hat{h}^\theta)}$$

$$p(h^{(t)} \mid h^\theta) = \pi_{h^{(t)}}$$



$$\sum_k = \frac{\sum_{k=1}^K \sum_{t=1}^T P(h^{(t)} = k | v^{(t)}, \hat{h}^\theta) (v^{(t)} - \mu_k) (v^{(t)} - \mu_k)^T}{\sum_{k=1}^K \sum_{t=1}^T P(h^{(t)} = k | v^{(t)}, \hat{h}^\theta)}$$



The algorithm -

The core of the EM in practice

- **INITIALIZATION:** set an initial $\{\mu_i^{(init)}, \Sigma_i^{(init)}, \pi_i^{(init)}\}$
- **STEP E:** Minimize $F(Q, P)$ with respect to $Q(h^{(t)})$ calculating for each possible value of $h^{(t)}$

$$Q(h^{(t)}) \leftarrow P(h^{(t)} | v^{(t)}, h^\theta)$$

EASY TO
COMPUTE !!!

for each t

- **STEP M:** Minimize $F(Q, P)$ with respect to h^θ , obtaining

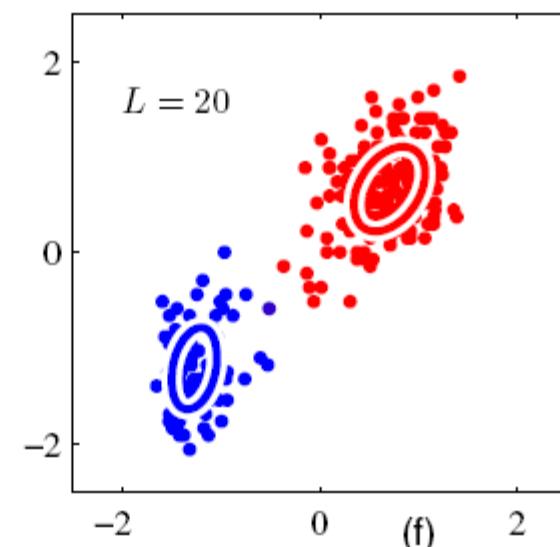
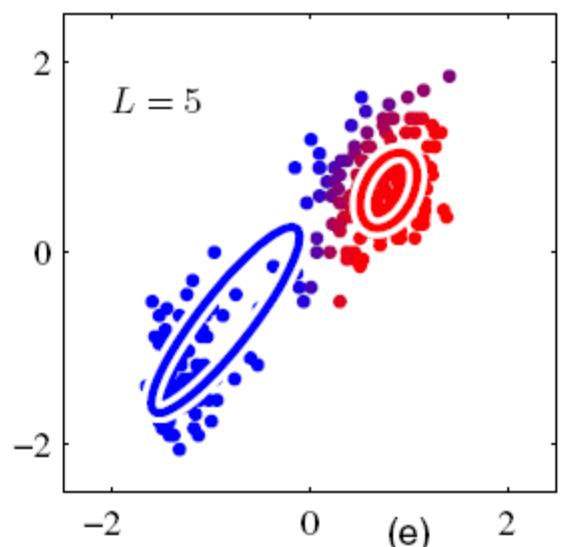
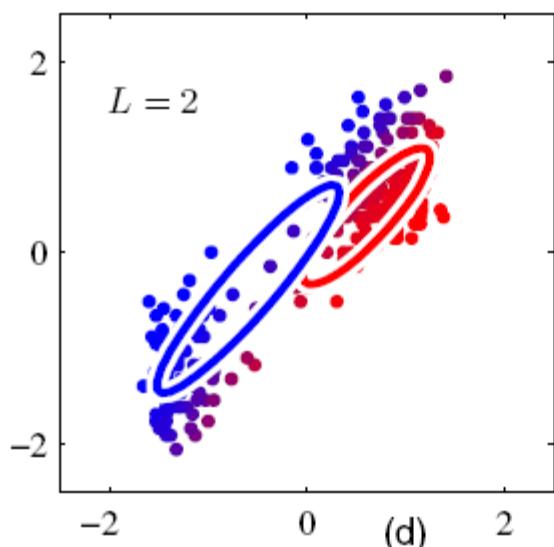
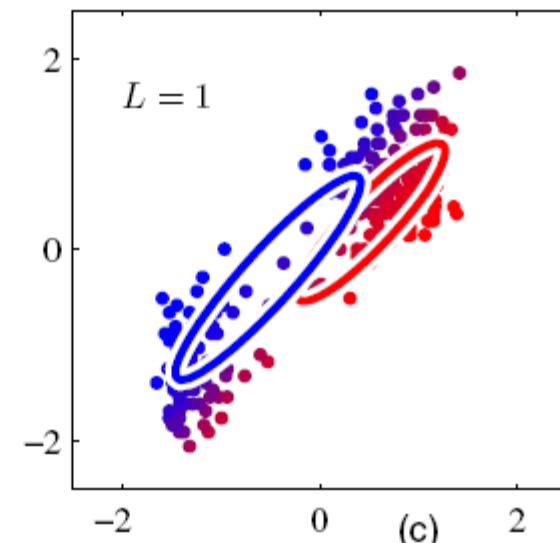
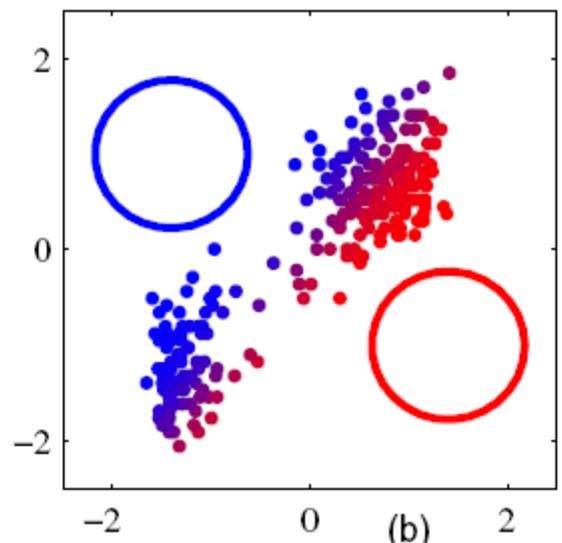
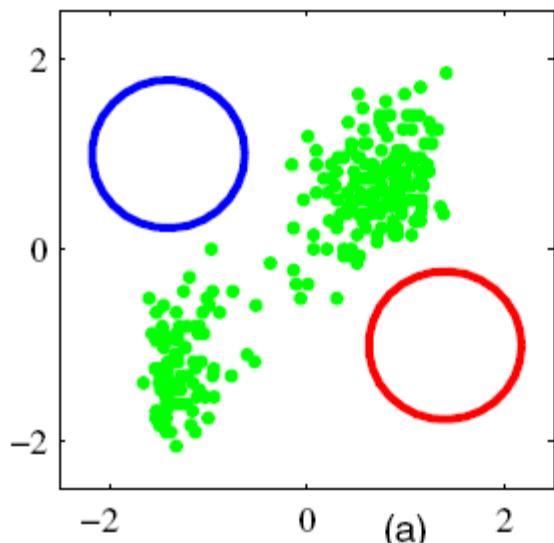
$$\{\mu_i^{(new)}, \Sigma_i^{(new)}, \pi_i^{(new)}\}$$

- **ITERATION** Loop until convergence!



Remarks

- The idea: introduce hidden variables which knowledge simplifies the computation of the parameters
- The hidden variables are related with the visible variables
- The decision of the hidden quantities is not an automatic process, and relies on the scientist
- In general, the EM will apply when we have to deal with mixtures



Mapping slide-libro

- Cap.3 Duda until 3.4.2
 - Maximum Likelihood Estimator for Variance is Biased: Proof (mat agg)
 - Dimostrazione di Jensen: <http://www.oilproject.org/lezione/disuguaglianza-di-jensen-e-aritmetico-geometrica-dimostrazione-2488.html>
- ⇒ Brendan J. Frey and Nebojsa Jojic. 2005. A Comparison of Algorithms for Inference and Learning in Probabilistic Graphical Models. *IEEE Trans. Pattern Anal. Mach. Intell.* 27, 9 (September 2005), 1392-1416.
DOI=10.1109/TPAMI.2005.169 (mat agg)
<http://dx.doi.org/10.1109/TPAMI.2005.169>
- ⇒ A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models (1998) by Jeff Bilmes